Streams

A stream is a connection to a source of data or to a destination for data

System.in is an input stream

System.out is an output stream

Import java.io

Open the stream > Use the stream > Close the stream

Openeing a stream

* There is data external to your program that you want to get, or you want to put data somewhere outside your program
* When you open a stream, you are making a connection to that external place

Example

* A file reader is used to connect to a file that will be used for input
* fileReader.read() method reads one character and returns it as an integer or -1 of there are no more characters

Manipulation

* A bufferedReader will convert the ints to characters, it can also read whole lines
* Constructor takes a FileReader parameter

Always close your streams!

Text files

* .txt are the simplest kind of files
* Compliers only work with text

Exeption

* Need to catch FileNotFoundException
  + Any block that catches IO exceptions also catches FileNotFound Exceptions

Serialization

* Writing/reading objects to files
* The class must be public
* Must implement Serializable interface
* The class must have a no-argument constructor
* All fields of the class must be serializable: either primitive types or serializable object types.

Serializable does not have any methos, simply tells java that this object is Serializable

ObjectOutputStream o = new ObjectOutputStream (new BufferedImputStream(new FileInput Stream()))

MyObject = (type)o.readObject();

o.close

File class

* Exists(): tests if file exists
* CanRead(): tests if the OS will let you read a file
* canWritte(): tests if the OS will let you write a file
* delete(): deletes the file, returns true if successful
* length(): returns the number of bytes in each file
* getName(): returns file name, except the preceding path.
* getPath(): returns the path name- the full name
* and many many more